CLAIMS:

What is claimed is:

1. A content addressable memory for finding data associated with key data comprising:

memory having a L level hierarchy for storing key entries in a sorted tree form where L >1 and storing data associated with the key entries;

an update block for inserting new key entries and associated data into the memory and deleting key entries and associated data from the memory while maintaining the sorted tree form of the key entries; and

at least one search block identifying a key entry and associated data that match the key data based on comparisons between the received key data and the key entries in the memory at each level of the hierarchy.

2. The memory according to claim 1, wherein the at least one search block comprises L search blocks, each search block corresponding to a memory level and wherein each search block returns an address value that identifies a subset of the next level of the memory for the search operation of the lower level search block until a matching entry or lack thereof is identified by the lowest level search block.

3. The memory according to claim 1, wherein the search blocks are arranged as a pipeline to receive new key data for search and retrieval operations in successive memory cycles.

4. The memory according to claim 1, wherein the at least one search block is configured to determine an exact match.

5. The memory according to claim 1, wherein the at least one search block is configured to determine a longest prefix match.

6. The memory according to claim 5, wherein the longest prefix match is determined based on masking the key data and key entries for different prefix lengths, determining when an exact match exists for different prefix lengths and selecting the longest exactly matching prefix.

7. The memory according to claim 1, wherein the memory includes rows of key entries arranged in order of key value.

8. The memory according to claim 7, wherein the memory includes rows of key entries arranged in cyclical ascending order.

9. The memory according to claim 7, wherein the memory includes rows of key entries arranged in cyclical descending order.

10. The memory according to claim 7, wherein the memory includes pointers to a maximum and a minimum value of each memory row.

11. The memory according to claim 10, wherein the memory stores a node value for each row that represents each row as a key entry in a higher level of the memory hierarchy.

12. The memory according to claim 11, wherein the node value is the maximum value for each row.

13. The memory according to claim 10, wherein the update block stores key entries displaced by insertion of a new key into the maximum value position of the next sequential row and updates the maximum and minimum value pointers of the next row based on the displacement.

14. The memory according to claim 10, wherein the update block stores key entries displaced by deletion of a key into the minimum value position of the next sequential row and updates the maximum and minimum value pointers of the next row based on the displacement.

15. The memory according to claim 13, wherein the memory includes an update value register and is implemented as a dual port random access memory that performs key entry insertion in three sequential operations performed in parallel.

16. The memory according to claim 14, wherein the memory includes an update value register is implemented as a dual port random access memory that performs key entry deletion in three sequential operations performed in parallel.

17. A method of finding data associated with key data in a content addressable memory comprising:

storing data in a memory having a L level hierarchy for storing key entries in a sorted tree form where L >1 and storing data associated with the key entries; and

searching each level of the L level memory hierarchy based on the key data; and

identifying a key entry and associated data that match the key data based on the searching at each level of the L level memory hierarchy.

18. The method according to claim 17, searching is performed as a pipeline for implementing search and retrieval operations in successive memory cycles.

19. The method according to claim 17, wherein the identifying identifies an exact match between the key data and one of the key entries.

20. The method according to claim 17, wherein the identifying identifies a longest prefix match between the key data and one of the key entries.

21. The method according to claim 20, wherein the longest prefix match is determined based on masking the key data and key entries for different prefix lengths, determining when an exact match exists for different prefix lengths and selecting the longest exactly matching prefix.

22. The method according to claim 17, wherein the memory includes rows of key entries arranged in order of key value.

23. The method according to claim 22, wherein the memory includes rows of key entries arranged in cyclical ascending order.

24. The method according to claim 22, wherein the memory includes rows of key entries arranged in cyclical descending order.

25. The method according to claim 22, wherein the memory includes pointers to a maximum and a minimum value of each memory row.

26. The method according to claim 25, wherein the memory stores a node value for each row that represents each row as a key entry in a higher level of the memory hierarchy.

27. The method according to claim 26, wherein the node value is the maximum value for each row.

28. The method according to claim 22, further comprising

inserting a new key entry and associated data into the memory, the insertion row being determined based on the value of the key entry;

moving a key entry displaced out of the insertion row into the maximum value position of the next sequential row; and

updating the maximum and minimum value pointers of the insertion row based on the displacement.

29. The method according to claim 22, further comprising

deleting a key entry and associated data from the memory at a deletion row;

moving a key entry from the next sequential row into the minimum value position of the deletion row; and

updating the maximum and minimum value pointers of the deletion row based on the displacement.

30. The method according to claim 28, wherein the memory includes an update value register and is implemented as a dual port random access memory that performs key entry insertion in three sequential operations performed in parallel.

31. The method according to claim 29, wherein the memory includes an update value register and is implemented as a dual port random access memory that performs key entry deletion in three sequential operations performed in parallel.